PAT-NO: JP406282448A

DOCUMENT-IDENTIFIER: JP 06282448 A

TITLE: SHARED RESOURCE EXCLUSIVE CONTROL SYSTEM

PUBN-DATE: October 7, 1994

INVENTOR-INFORMATION:
NAME
SUZUKI, YOSHINAO

ASSIGNEE-INFORMATION:
NAME COUNTRY
NEC CORP N/A

APPL-NO: JP05093656

APPL-DATE: March 29, 1993

INT-CL (IPC): G06F009/46

ABSTRACT:

PURPOSE: To shorten the lock wait time of a following lock waiting task and to improve the availability of a shared resource.

CONSTITUTION: A lock request processing means 1 process a lock request to call a lock order change means 3 and a lock priority setting means 4. An unlock request processing means 2 processes an unlock request to call a lock priority setting means 4. The lock priority setting means 4 sets weight for each resource in advance and sets a value, calculated from a function f(x) (function which makes f(a)≥f(b) hold when a>b and includes (c) and (d) so that c>d satisfying f(c)>f(d)) of the sum of the weight of resources where tasks in lock wait states are present among locked resources of tasks, as the lock priority of a task. The lock order change means 3 rearranges a lock wait queue so that the lock right is granted to the lock waiting tasks in the decreasing order of the lock priority.

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平6－282448

(43)公開日　平成６年(1994)10月７日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/46	３４０	Ｆ　8120－5B		

審査請求　有　　請求項の数３　ＦＤ　（全　８　頁）

(21)出願番号　特願平5－93656

(22)出願日　平成５年(1993)３月29日

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号

(72)発明者　鈴木　善尚
東京都港区芝五丁目７番１号　日本電気株式会社内

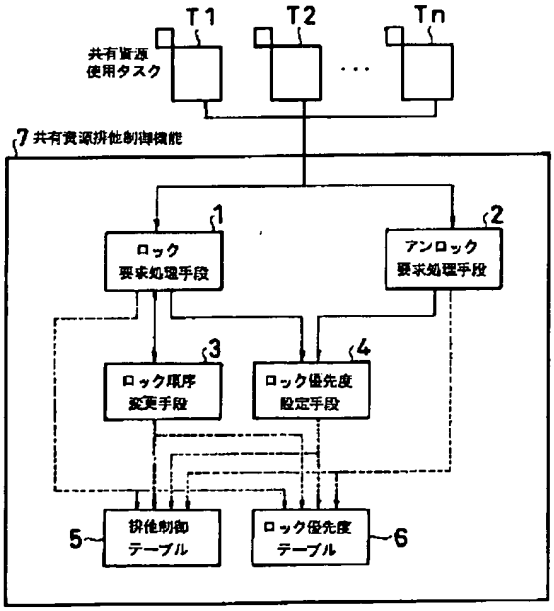
(74)代理人　弁理士　河原　純一

(54)【発明の名称】　共有資源排他制御方式

(57)【要約】

【目的】　後続のロック待ちタスクのロック待ち時間を減少させるとともに共有資源の使用効率を向上させる。

【構成】　ロック要求処理手段１は、ロック要求を処理してロック順序変更手段３およびロック優先度設定手段４を呼び出す。アンロック要求処理手段２は、アンロック要求を処理してロック優先度設定手段４を呼び出す。ロック優先度設定手段４は、あらかじめ各資源に重みを設定しておき、タスクのロック中資源のうちのロック待ちしているタスクが存在する資源の重みの和ｘの関数ｆ（ｘ）（ａ＞ｂならばｆ（ａ）≧ｆ（ｂ）が成立し、かつｆ（ｃ）＞ｆ（ｄ）を満たすｃ＞ｄなるｃおよびｄが存在する関数）によって算出された値を、該タスクのロック優先度として設定する。ロック順序変更手段３は、ロック待ちタスクへのロック権付与順序がロック優先度の高い順となるようロック待ち行列を並べ換える。



1

【特許請求の範囲】

【請求項1】　計算機システムの共有資源排他制御方式において、

あらかじめ各資源に重みを設定しておき、タスクのロック中資源のうちのロック待ちしているタスクが存在する資源の重みの和xの関数f（x）（a＞bならばf（a）≧f（b）が成立し、かつf（c）＞f（d）を満たすc＞dなるcおよびdが存在する関数）によって算出された値を、該タスクのロック優先度として設定するロック優先度設定手段と、

ロック待ちタスクへのロック権付与順序がロック優先度の高い順となるようロック待ち行列を並べ換えるロック順序変更手段とを有することを特徴とする共有資源排他制御方式。

【請求項2】　各資源の重みを1に固定したことを特徴とする請求項1記載の共有資源排他制御方式。

【請求項3】　前記関数をf（x）＝xとしたことを特徴とする請求項1記載の共有資源排他制御方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は共有資源排他制御方式に関し、特に計算機システムの共有資源排他制御方式に関する。

【0002】

【従来の技術】計算機システムにおいて複数のタスクから使用される資源については、それへのアクセスを逐次化するために排他制御が必要である。このような排他制御のために、計算機システムには、ロック／アンロック（あるいはENQ／DEQ）と呼ばれる基本機能が備えられており、ロック要求には通常排他モードと共有モードとの2種類がある。

【0003】排他制御の基本的動作は、次のようなものである。あるタスクがロック要求を出したとき、ロック要求対象資源が他のタスクによりロックされていない場合、または、その資源のロックモードと該タスクのロック要求のモードとがともに共有モードの場合、すぐにロック権が付与される。しかし、その資源のロックモード、または該タスクのロック要求のモードが排他モードの場合には、ロック待ちとなる。

【0004】また、資源をロックしていたタスクがアンロック要求を出したときは、該タスクのロック権の行使が終了する。他にロックしているタスクがなく、かつロック待ちタスクがあり、それらのうちの最も早くロック要求したタスクの要求モードが排他モードの場合は、該タスクのみにロック権を付与し、共有モードならば現在共有モードでロック待ちしているタスクすべてにロック権を付与する。

【0005】従来技術の文献としては、例えば特開平2－118843号公報，特開平3－83142号公報等がある。

2

【0006】

【発明が解決しようとする課題】上述した従来の共有資源排他制御方式では、複数の資源を同時に使用して処理するときに各資源に対し逐次ロック要求するが、いくつかの資源のロックが成功し、次の資源をロックしようとしてロック待ちとなった場合、すでにロックした資源に対してロック待ちしている他のタスクの待ち時間が不必要に長くなり、資源の使用効率が低下するという欠点がある。

【0007】本発明の目的は、上述の点に鑑み、後続のロック待ちタスクのロック待ち時間を減少させるとともに共有資源の使用効率を向上させるようにした共有資源排他制御方式を提供することにある。

【0008】

【課題を解決するための手段】本発明の共有資源排他制御方式は、計算機システムの共有資源排他制御方式において、あらかじめ各資源に重みを設定しておき、タスクのロック中資源のうちのロック待ちしているタスクが存在する資源の重みの和xの関数f（x）（a＞bならばf（a）≧f（b）が成立し、かつf（c）＞f（d）を満たすc＞dなるcおよびdが存在する関数）によって算出された値を、該タスクのロック優先度として設定するロック優先度設定手段と、ロック待ちタスクへのロック権付与順序がロック優先度の高い順となるようロック待ち行列を並べ換えるロック順序変更手段とを有する。

【0009】

【実施例】次に、本発明について図面を参照して詳細に説明する。

【0010】図1は、本発明の一実施例に係る共有資源排他制御方式の構成を示すブロック図である。本実施例の共有資源排他制御方式を実現する共有資源排他制御機能7は、ロック要求の処理を行うロック要求処理手段1と、アンロック要求の処理を行うアンロック要求処理手段2と、ロック優先度の高い順にロック待ち行列を並べ換えるロック順序変更手段3と、タスクのロック中資源のうちのロック待ちしているタスクが存在する資源の重みの和の関数で算出される値を該タスクのロック優先度として設定するロック優先度設定手段4と、排他制御テーブル5と、ロック優先度テーブル6とから構成されている。なお、図1中、符号T1～Tn（nは正整数）は、共有資源使用タスク（以下、単にタスクという）を示す。

【0011】排他制御テーブル5は、図2に示すように、計算機システム内の排他制御対象資源を一意に識別するための名前である資源名をキーとする。各資源に対するエントリは、資源名部と、重み部と、ロックモード部と、ロック中タスクリストと、ロック待ち行列とからなる。重み部は、その資源の重みを記録する。ロックモード部は、資源がロックされているときのロックモード

を記録する。ロック中タスクリストは、その資源をロックしているタスクのタスク識別子を記録する。ロック待ち行列は、ロック待ち状態にあるロック要求の待ち行列であり、これらのロック要求を出しているタスクのタスク識別子とロックモードとの組を記録する。

【００１２】ロック優先度テーブル６は、図２に示すように、計算機システム内のタスクを一意に識別するための名前であるタスク識別子をキーとする。各タスクに対するエントリは、タスク識別子部と、ロック待ち資源名部と、重み総和部と、ロック優先度部とからなる。ロック待ち資源名部は、そのタスクがロック待ちしている資源の資源名を記録する。重み総和部は、そのタスクがロックしている資源のうちのロック待ちしているタスクが存在している資源の重みの和を記録する。ロック優先度部は、そのタスクのロック優先度を記録する。

【００１３】ロック要求処理手段１は、下記（１）または（２）のロック要求の処理を行う。

【００１４】（１）　ロック要求の対象資源に対応する排他制御テーブル５のエントリのロックモード部（以下、特に断らない限り、単に重み部，ロックモード部，ロック中タスクリストおよびロック待ち行列と呼んだときには、ロック要求またはアンロック要求の対象資源に対応する排他制御テーブル５のエントリを考える）が空の場合、またはロックモード部と該ロック要求のロックモードとがともに共有モードの場合は、ロック要求処理手段１は、ロックモード部に該ロック要求のロックモードを記録し、ロック中タスクリストに該ロック要求を出したタスクのタスク識別子を追加する。次に、ロック要求処理手段１は、ロック待ち行列が空でなければ、該ロック要求の対象資源の資源名と該ロック要求を出したタスクのタスク識別子と優先度増加要求の旨とをパラメータとしてロック優先度設定手段４を呼び出す。最後に、ロック要求処理手段１は、該ロック要求を出したタスクの処理を中断させる。

【００１５】（２）　ロックモード部が空でなくかつ該ロック要求のロックモードが排他モードの場合、またはロックモード部が排他モードの場合は、ロック要求処理手段１は、該ロック要求を出したタスクのタスク識別子とロックモードとの組をロック待ち行列の最後に追加し、該タスクのタスク識別子に対応するロック優先度テーブル６のロック待ち資源名部（以下、特に断らない限り、単にロック待ち資源名部，重み総和部およびロック優先度部と呼んだときには、タスクに対応するロック優先度テーブル６のエントリを考える）に該ロック要求の対象資源の資源名を記録し、該資源名をパラメータとしてロック順序変更手段３を呼び出す。次に、ロック要求処理手段１は、ロック待ち行列に記録されている要求の数が１ならば、ロック中タスクリストに記録されている各タスクに対し、そのタスクのタスク識別子と該ロック要求の対象資源の資源名と優先度増加要求の旨とをパラメータとしてロック優先度設定手段４を呼び出し、そのタスクに対応するロック優先度テーブル６のエントリのロック待ち資源名部が空でない場合は、該ロック待ち資源名部の資源名をパラメータとしてロック順序変更手段３を呼び出す。最後に、ロック要求処理手段１は、該ロック要求を出したタスクの処理を中断させる。

【００１６】アンロック要求処理手段２は、アンロック要求の処理を行い、該アンロック要求を出したタスクのタスク識別子をロック中タスクリストから削除し、ロック待ち行列が空でない場合は、該アンロック要求を出したタスクのタスク識別子と該アンロック要求の対象資源の資源名と優先度減少要求の旨とをパラメータとしてロック優先度設定手段４を呼び出す。次に、アンロック要求処理手段２は、ロック中タスクリストが空となっている場合は、ロックモード部を消去する。さらに、ロック待ちタスクがあれば、アンロック要求処理手段２は、①次のロック待ちタスクが排他モードでロック要求している場合は、ロックモード部に排他モードを、ロック中タスクリストに該タスクのタスク識別子をそれぞれ記録し、該タスクのロック要求をロック待ち行列から削除し、②次のロック待ちタスクが共有モードでロック要求している場合は、ロックモード部に共有モードを記録し、ロック待ち行列内の共有モードでロック要求を出しているすべてのタスクのタスク識別子をロック中タスクリストに記録し、ロック待ち行列内のすべての共有モードのロック要求を削除する。

【００１７】続いて、アンロック要求処理手段２は、前記①または②でロック中タスクリストに記録したすべてのタスク識別子について、対応するロック優先度テーブル６のエントリのロック待ち資源名部に空を記録する。次いで、アンロック要求処理手段２は、ロック待ち行列が空でなければ、ロック中タスクリストに記録したすべてのタスク識別子について、各タスク識別子と該アンロック要求の対象資源の資源名と優先度増加要求の旨とをパラメータとしてロック優先度設定手段４をタスク数分だけ繰り返し呼び出す。次いで、アンロック要求処理手段２は、これらのタスクの処理を再開させる。

【００１８】ロック順序変更手段３は、パラメータで指定された資源名に対応する排他制御テーブル５のエントリに関し、該エントリのロック待ち行列を、ロック優先度テーブル６のロック優先度部を参照して、各ロック要求を出したタスクのロック優先度の高い順に並べ換える。このとき、ロック優先度の等しいタスクのロック要求は先着順に並べる。

【００１９】ロック優先度設定手段４は、パラメータで指定された要求が優先度増加要求ならば、パラメータで指定された資源名に対応する排他制御テーブル５のエントリの重み部をパラメータで指定されたタスク識別子に対応するロック優先度テーブル６のエントリの重み総和部に加算し、優先度減少要求ならば、パラメータで指定

された資源名に対応する排他制御テーブル5のエントリの重み部をパラメータで指定されたタスク識別子に対応するロック優先度テーブル6のエントリの重み総和部から減算する。次に、ロック優先度設定手段4は、パラメータで指定されたタスク識別子に対応するロック優先度テーブル6のエントリに関し、重み総和部を参照してロック優先度を算出し、この値を該エントリのロック優先度部に記録する。ロック優先度の算出は、重み総和xの関数f（x）（a＞bならばf（a）≧f（b）が成立し、かつf（c）＞f（d）を満たすc＞dなるcおよびdが存在する関数）を用いる。

【0020】図3は、4つのタスクT1～T4が2つの資源R1およびR2に対して順次ロック要求およびアンロック要求を行う状況を示すタイミングチャートである。

【0021】図4（a）～（d）は、図3の各タイミングta～tdでの排他制御テーブル5およびロック優先度テーブル6の状態を示している。なお、資源の重みはR1に2、R2に1があらかじめ設定されているものとし、時刻tより前では資源R1およびR2はロックされていないものとする。また、関数f（x）として、f（0）＜f（1）＜f（2）＜f（3）が成立するような関数が用いられ、ロック優先度テーブル6のロック待ち資源名部，重み総和部およびロック優先度部がそれぞれ空，0およびf（0）に初期化されているものとする。

【0022】次に、このように構成された本実施例の共有資源排他制御方式の動作について、図3および図4を参照して具体的に説明する。

【0023】時刻tでタスクT1が資源R1を排他モードでロック要求すると、ロック要求処理手段1が呼び出される。ロック要求処理手段1は、資源R1がロックされていないので、該タスクT1にロック権を付与し、ロックモード部に排他モードを、ロック中タスクリストにタスク識別子T1をそれぞれ記録する。次に、ロック要求処理手段1は、ロック待ち行列が空なので、ロック優先度設定手段4を呼び出すことなく、該ロック要求の処理を完了する。

【0024】時刻t＋1でタスクT2が資源R1を排他モードでロック要求すると、ロック要求処理手段1が呼び出される。ロック要求処理手段1は、資源R1がすでに排他モードでロックされているので、該タスクT2をロック待ちとし、タスク識別子T2と排他モードとの組をロック待ち行列に記録する。次に、ロック要求処理手段1は、タスク識別子T2に対応するロック待ち資源名部に資源名R1を記録し、該資源の資源名R1をパラメータとしてロック順序変更手段3を呼び出す。ロック順序変更手段3は、ロック待ち行列の並べ換えを行うが、ロック待ちタスクは1つなので結果は変わらない。続いて、ロック要求処理手段1は、ロック待ち行列内のロック要求の数が1なので、ロック中タスクリストのタスク識別子T1と該ロック要求の対象資源の資源名R1と優先度増加要求の旨とをパラメータとしてロック優先度設定手段4を呼び出す。ロック優先度設定手段4は、タスクT1に対応するロック優先度テーブル6のエントリの重み総和部に資源名R1に対応する排他制御テーブル5のエントリの重み部の値2を加算し、タスクT1に対応するロック優先度テーブル6のエントリのロック優先度部にf（2）を設定する。続いて、ロック要求処理手段1は、タスクT1に対応するロック優先度テーブル6のエントリのロック待ち資源名部が空なので、ロック順序変更手段3を呼び出すことなく、タスクT2の処理を中断させる。

【0025】時刻t＋2でタスクT3が資源R2を共有モードで、時刻t＋3でタスクT4が資源R2を排他モードでそれぞれロック要求すると、タスクT1およびタスクT2の場合と同様に処理が行われ、タスクT3にロック権が付与され、タスクT4はロック待ちとなる。この結果、時刻taでの排他制御テーブル5およびロック優先度テーブル6は、図4（a）の状態となる。

【0026】時刻t＋4でタスクT3が資源R1を共有モードでロック要求すると、ロック要求処理手段1が呼び出される。ロック要求処理手段1は、資源R1がすでに排他モードでロックされているので、該タスクのタスク識別子T3と共有モードとの組を資源名部R1に対応するロック待ち行列の最後に追加する。次に、ロック要求処理手段1は、ロック待ち資源名部に資源R1の資源名を記録し、該資源の資源名R1をパラメータとしてロック順序変更手段3を呼び出す。ロック順序変更手段3は、タスクT2およびタスクT3のロック優先度がそれぞれf（0）およびf（1）であり、f（0）＜f（1）よりタスクT3のロック要求、タスクT2のロック要求の順にロック待ち行列を並べ換える。すなわち、先にロック要求したタスクT2をタスクT3が追い越し、タスクT3へのロック権付与が優先される。続いて、ロック要求処理手段1は、ロック待ち行列内のロック要求の数が1ではないため、ロック優先度設定手段4を呼び出すことなく、タスクT3の処理を中断させる。この結果、時刻tbでの排他制御テーブル5およびロック優先度テーブル6は、図4（b）の状態となる。

【0027】時刻t＋5でタスクT1が資源R1をアンロック要求すると、アンロック要求処理手段2が呼び出される。アンロック要求処理手段2は、ロック中タスクリストからタスク識別子T1を削除する。次に、アンロック要求処理手段2は、ロック待ち行列が空でないので、該タスクのタスク識別子T1と該アンロック要求の対象資源の資源名R1と優先度減少要求の旨とをパラメータとしてロック優先度設定手段4を呼び出す。ロック優先度設定手段4は、タスク識別子T1に対応するロック優先度テーブル6のエントリの重み総和部から資源名

Ｒ１に対応する排他制御テーブル５のエントリの重み部の値２を減じ、タスクＴ１に対応するロック優先度テーブル６のエントリのロック優先度部にｆ（０）を設定する。この結果、ロック中タスクリストが空となるので、アンロック要求処理手段２は、ロックモード部を消去する。さらに、アンロック要求処理手段２は、ロック待ち行列が空ではないので、最初のロック要求のタスクＴ３にロック権を付与し、ロックモード部に共有モードを、ロック中タスクリストにタスク識別子Ｔ３をそれぞれ記録する。次に、アンロック要求処理手段２は、該タスクＴ３に対応するロック優先度テーブル６のロック待ち資源名部を空とし、ロック待ち行列が空でないので、タスク識別子Ｔ３と該ロック要求の資源名Ｒ１と優先度増加要求の旨とをパラメータとしてロック優先度設定手段４を呼び出す。ロック優先度設定手段４は、タスクＴ３に対応するロック優先度テーブル６のエントリの重み総和部に資源名Ｒ１に対応する排他制御テーブル５のエントリの重み部の値２を加え、タスクＴ３に対応するロック優先度テーブル６のロック優先度部にｆ（３）を設定した後、タスクＴ３の処理を再開させる。この結果、時刻ｔｃでの排他制御テーブル５およびロック優先度テーブル６は、図４（ｃ）の状態となる。

【００２８】時刻ｔ＋６でタスクＴ３が資源Ｒ１およびＲ２を続けてアンロック要求すると、タスクＴ１によるアンロック要求と同様に処理が行われ、資源Ｒ１においてはタスクＴ２に、資源Ｒ２においてはタスクＴ４にそれぞれロック権が付与される。この結果、時刻ｔｄでの排他制御テーブル５およびロック優先度テーブル６は、図４（ｄ）の状態となる。

【００２９】なお、上記実施例では、関数ｆ（ｘ）がｆ（０）＜ｆ（１）＜ｆ（２）＜ｆ（３）を満足する関数であると仮定したが、ｆ（ａ）＝ｆ（ｂ）（ａ＜ｂ）のような場合を含む関数も用いることができる。この関数は、重み総和に従って厳密にロック優先度を制御するのではなく、重み総和のある範囲に対し１つのロック優先度を対応させることになり、重み総和が大差ないときには先着順にロック権を付与した方がよい場合に向いている。このような関数を用いた場合、動作としては、ロック順序変更手段３でのロック待ち行列の並べ換え時に重み総和ｂのタスクが重み総和ａのタスクに優先されずに先着順に並べられる点が異なる以外は同様の処理となる。

【００３０】また、上記実施例では、各資源の重みを異ならしめた場合について説明したが、各資源の重みをすべて１に固定するようにしてもよい。このようにすると、各資源のロック待ち行列長の時間平均値や１回当たりのロック時間平均値などが大差なく、どの資源も同等に扱うことができる場合に好適である。また、排他制御テーブル５の重み部の値をロック優先度テーブル６の重み総和部に加減算する処理を、重み総和部に定数１を加減算する処理に置き換えることができる。これにより、排他制御テーブル５の重み部を削除することができ、排他制御テーブル５に必要なメモリ量を削減することができる。動作は、重み総和部への加減算処理が、前記処理になる以外は、上記実施例の場合と同様である。

【００３１】さらに、上記実施例では、ロック優先度を算出する関数ｆ（ｘ）を使用する場合について説明したが、ロック優先度を算出する関数をｆ（ｘ）＝ｘとし、重み総和そのものをロック優先度とするようにしてもよい。このようにすれば、重み総和による厳密な優先度制御を必要とする場合に好適である。また、排他制御テーブル５の重み部をロック優先度テーブル６の重み総和部に加減算する処理を、ロック優先度部に直接加減算するように変更することにより、ロック優先度テーブル６の重み総和部を削除することができ、ロック優先度テーブル６に必要なメモリ量とロック優先度算出に必要な実行ステップ数とを削減することができる。動作は、ロック優先度算出処理が前記処理になる以外は、上記実施例の場合と同様である。

【００３２】

【発明の効果】以上説明したように本発明は、ロック待ちが発生している資源の持つ重み総和の増加にともなって増加するようなロック優先度を設け、ロック優先度の高いタスクに対しロック優先度の低いタスクに優先してロック権が付与されるように制御することにより、他のタスクを待たせているタスクが新たな資源をロック要求してロック待ちとなった場合にロック待ち時間を減少させることができ、該タスクがロックしている資源に対しロック待ちしている他のタスクのロック待ち時間を減少させるとともに、共有資源の使用効率を向上させる効果がある。

【図面の簡単な説明】

【図１】本発明の一実施例に係る共有資源排他制御方式の構成を示すブロック図である。

【図２】本実施例の共有資源排他制御方式における排他制御テーブルおよびロック優先度テーブルの構成を示す図である。

【図３】本実施例の共有資源排他制御方式の動作を説明するためのロック／アンロックタイミングチャートである。

【図４】本実施例の共有資源排他制御方式の動作を説明するための排他制御テーブルおよびロック優先度テーブルの状態を示す図であり、（ａ）は時刻ｔａにおける排他制御テーブルおよびロック優先度テーブルの状態を示し、（ｂ）は時刻ｔｂにおける排他制御テーブルおよびロック優先度テーブルの状態を示し、（ｃ）は時刻ｔｃにおける排他制御テーブルおよびロック優先度テーブルの状態を示し、（ｄ）は時刻ｔｄにおける排他制御テーブルおよびロック優先度テーブルの状態を示す。

【符号の説明】

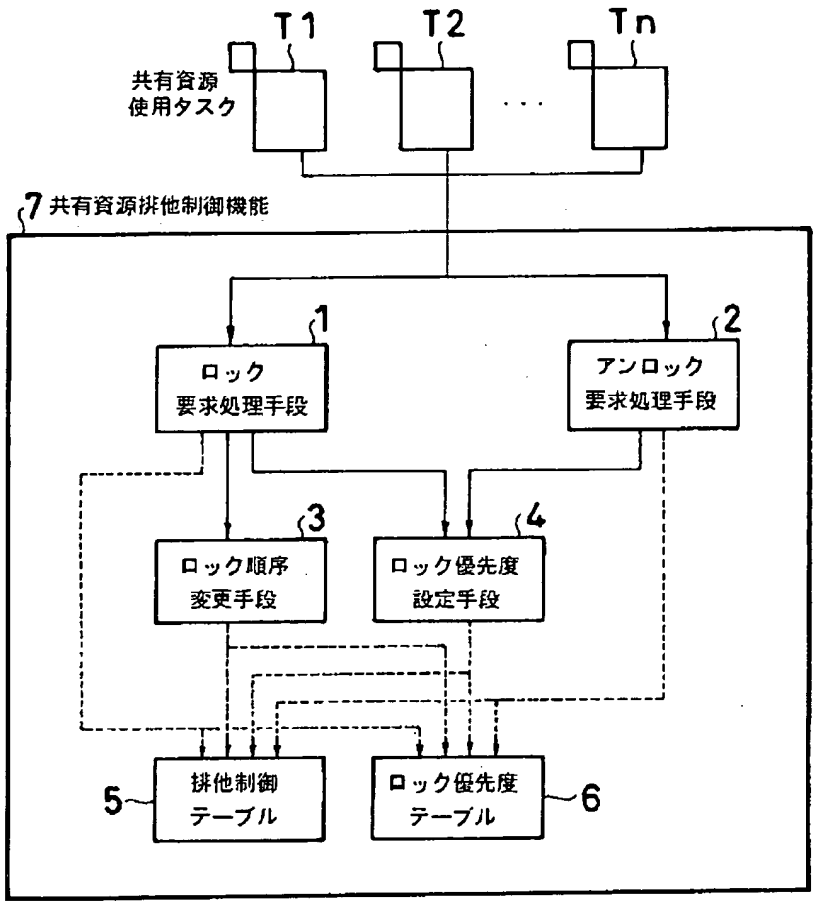
9

1　ロック要求処理手段
2　アンロック要求処理手段
3　ロック順序変更手段
4　ロック優先度設定手段
5　排他制御テーブル

10

6　ロック優先度テーブル
7　共有資源排他制御機能
T1～Tn　タスク
R1，R2　資源

【図1】



【図2】

5 排他制御テーブル

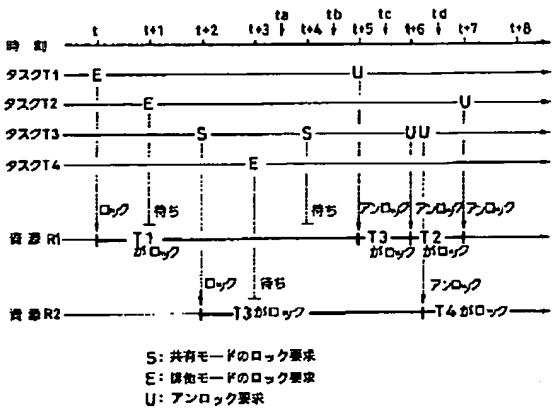
資源名部	重み部	ロックモード部	ロック中タスクリスト	ロック待ち行列
R1	1	S	T1	(T2,E)

6 ロック優先度テーブル

タスク識別子部	ロック待ち資源名部	重み総和部	ロック優先度部
T1		1	f(1)
T2	R1	0	f(0)

Rn：資源名　　S：共有モード　　f(x)：ロック優先度（x＝重み総和）
Tn：タスク識別子　　E：排他モード

【図3】



【図4】

(a)時刻ｔａ

排他制御テーブル 5

資源名部	重み部	ロックモード部	ロック中タスクリスト	ロック待ち行列
R1	2	E	T1	(T2,E)
R2	1	S	T3	(T4,E)

ロック優先度テーブル 6

タスク識別子部	ロック待ち資源名部	重み総和部	ロック優先度部
T1		2	f(2)
T2	R1	0	f(0)
T3		1	f(1)
T4	R2	0	f(0)

(b)時刻ｔｂ

排他制御テーブル 5

資源名部	重み部	ロックモード部	ロック中タスクリスト	ロック待ち行列
R1	2	E	T1	(T3,S)(T2,E)
R2	1	S	T3	(T4,E)

ロック優先度テーブル 6

タスク識別子部	ロック待ち資源名部	重み総和部	ロック優先度部
T1		2	f(2)
T2	R1	0	f(0)
T3	R1	1	f(1)
T4	R2	0	f(0)

(c)時刻ｔｃ

排他制御テーブル 5

資源名部	重み部	ロックモード部	ロック中タスクリスト	ロック待ち行列
R1	2	S	T3	(T2,E)
R2	1	S	T3	(T4,E)

ロック優先度テーブル 6

タスク識別子部	ロック待ち資源名部	重み総和部	ロック優先度部
T1		0	f(0)
T2	R1	0	f(0)
T3		3	f(3)
T4	R2	0	f(0)

(d)時刻ｔｄ

排他制御テーブル 5

資源名部	重み部	ロックモード部	ロック中タスクリスト	ロック待ち行列
R1	2	E	T2	
R2	1	E	T4	

ロック優先度テーブル 6

タスク識別子部	ロック待ち資源名部	重み総和部	ロック優先度部
T1		0	f(0)
T2		0	f(0)
T3		0	f(0)
T4		0	f(0)